# Part 8: Adding a class

## Exercise Instructions

We're well on the way now! This one's quite easy. Whereas you might be expecting something to do with inheritence or polymorphism, there's no point using a feature unless you need it - and we don't need it here. So for this exercise, you just have to pull the calculation logic and the data points used into their own class.

Good luck!

### Step 1: Update the Code

First off, we create a new class where the calculation code will live:

- Create a new class `LoanCalculator` (right-click on the package and then do **New > Java Class**, selecting **Class** as the type).
- Add 3 fields which hold the values for the calculation to the class (Hint: these are the same as the input variables)
- Add a constructor to the class which takes all 3 fields as parameters. (To do this, right-click somewhere in a blank part of the code under the field declarations and then select **Generate... > Constructor**.)
- Add getter methods to the class for the fields (again, right-click somewhere in a blank part of the code under the constructor now and then select **Generate... > Getter**.

Next, we migrate the calculation logic into this class (so the logic lives with the data it operates on):

- Cut the method `calculateRepaymentAmount` from the `App` class and paste it into the `LoanCalculator` class, changing it from `static` to non-static (just remove the keyword to do this).

Finally, we need to use the class in our main class:

- Change the code as below to use the class - this should now work as is:

**old code snippet**

```
printResult(calculateRepaymentAmount(amount, years, interestRate));
```

**new code snippet**

```
LoanCalculator calculator = new LoanCalculator(amount, years, interestRate);
double repaymentAmount = calculator.calculateRepaymentAmount(amount, years, intere
stRate);

printResult(repaymentAmount);
```

- Bonus: can you think of a way to tighten up this new code snippet? (Hint: we did it in the last challenge!)

## Step 2: Run the Code

- Run the code to make sure everything works as before.

You might also like to debug the code so you can see the `LoanCalculator` being created and step through the various methods along the execution of the program to see how al that hangs together too firsthand.